

### Printing Tokens

จงเขียนโปรแกรมเพื่อพิมพ์คำแต่ละคำในประโยค

ข้อมูลนำเข้า      ข้อความ ความยาวไม่เกิน 100 อักขระ

ข้อมูลนำออก      แต่ละบรรทัดพิมพ์คำหนึ่งคำ

ตัวอย่าง

| Input                | Output                        |
|----------------------|-------------------------------|
| Today is Monday.     | Today<br>is<br>Monday.        |
| C programming is fun | C<br>programming<br>is<br>fun |

ประโยค 01 เรขคณิต 100

```
if (str == '\n') {
    printf("%s\n", str);
}
```

### ข้อความแพนแกรม (Pangram String)

pangram คือข้อความที่มีอักขระภาษาอังกฤษ a-z ทุกอักขระปรากฏในข้อความ กำหนดให้อักขระพิมพ์เล็ก หรือพิมพ์ใหญ่ ถือว่าปรากฏในข้อความเหมือนกัน เช่น w หรือ W

จงเขียนโปรแกรมเพื่อตรวจสอบข้อความว่าเป็น pangram string หรือไม่

ข้อมูลนำเข้า      ข้อความ ความยาวไม่เกิน 100 อักขระ

ข้อมูลนำออก      พิมพ์คำว่า pangram หรือ not pangram ตามเงื่อนไขที่กำหนด

ตัวอย่าง

```
int check[26] = {0};
```

| Input  | Output      |
|--|-------------|
| We promptly judged antique ivory buckles for the next prize                | pangram     |
| We promptly judged antique ivory buckles for the prize                     | not pangram |
| The string contains all letters in the English alphabet, so return pangram | not pangram |

int check[26] = {0};  
2++      ถ้าไม่ครบ ก็ not pangram

## printing Token

C++

```
1 #include <bits/stdc++.h>
2 using namespace std;
3
4 int main(){
5     string s;
6     getline(cin, s);
7
8     for(int i = 0; i < s.size(); i++){
9
10        if(s[i] == ' '){
11            cout << endl;
12        }else{
13            cout << s[i];
14        }
15    }
16    return 0;
17 }
18 }
```

C

```
1 #include <stdio.h>
2 #include <string.h>
3
4 int main(){
5     int n;
6     char str[100];
7     gets(str);
8     n = strlen(str);
9
10    for(int i = 0; i < n; i++){
11        if(str[i] == ' '){
12            printf("\n");
13        }else{
14            printf("%c", str[i]);
15        }
16    }
17    return 0;
18 }
```

## Pangram string

```
1 #include <stdio.h>
2 #include <string.h>
3
4 int main(){
5     char str[101];
6     gets(str);
7     int s = strlen(str); {0=1, 0, }
8     int check[26] = {0};
9     int m = 65, n = 97;
10
11    for(int i = 0; i < s; i++){
12        if((str[i] >= 'A' && (str[i] <= 'Z'))){
13            check[str[i] % m]++;
14        }else if((str[i] >= 'a' && (str[i] <= 'z'))){
15            check[str[i] % n]++;
16        }
17    }
18    int pangram = 1;
19    for(int i = 0; i < 26; i++){
20        if(check[i] == 0){
21            pangram = 0;
22            break;
23        }
24    }
25    if(pangram == 1){
26        printf("pangram");
27    }else{
28        printf("not pangram");
29    }
30
31    return 0;
32 }
```

**Palindrome**

พาลินโดรม (palindrome) คือ คำที่ใช้มีลำดับของอักขระเรียงกันจากซ้ายไปขวาและขวาไปซ้ายมายังตำแหน่งกึ่งกลางของคำนั้น อยู่ในลักษณะสมมาตรกัน ตัวอย่างของพาลินโดรมได้แก่ ABA, ABBA, ABAABA, ABABABA

จงเขียนโปรแกรม เพื่ออ่านข้อมูลนำเข้าจากแป้นพิมพ์ (Standard input) และแสดงผลลัพธ์ออกทางจอภาพ (Standard Output) โดยข้อมูลนำเข้าเป็นคำหนึ่งคำให้บอกว่าคำนั้นเป็น พาลินโดรม หรือ ไม่เป็นพาลินโดรม โดย ถ้าเป็นพาลินโดรม ให้แสดงคำว่า Yes และถ้าไม่เป็นทั้งสองแบบให้แสดงคำว่า No

ข้อมูลนำเข้า บรรทัดที่ 1 จำนวนชุดทดสอบ T โดยที่  $2 \leq T \leq 10$

บรรทัดที่ 2 ถึง T + 1 เป็นคำที่มีความยาว n โดยที่  $2 \leq n \leq 200$

ข้อมูลนำออก ผลการตรวจสอบ Palindrome จำนวน T บรรทัด

ตัวอย่าง *ดู 1 บรรทัด*  
*เลข +1 บรรทัด -1*

| Input   | Output |
|---------|--------|
| 4       | Yes    |
| ABAABA  | Yes    |
| anna    | Yes    |
| redder  | Yes    |
| madam   | Yes    |
| 3       | No     |
| elbow   | Yes    |
| stats   | Yes    |
| rotator | Yes    |

\*\*\*หมายเหตุ อักขระพิมพ์เล็กและอักขระพิมพ์ใหญ่ เช่น a A ถือว่าเหมือนกัน

*นำมา*

*รวมผล*

*arr = ABA*

*arr (รีเวิร์ด) = ABA*

*มาเช็คทีละตัวว่าเหมือนกัน*



03\_Palindrom (Kamal).cpp

```
#include <iostream>
#include <string>
using namespace std;

int n;
string str1[10];
string str2[10];

string check(string s1, string s2){
    if(s1 == s2){
        return "Yes";
    }else{
        return "No";
    }
}

int main(){
    cin >> n;
    for(int i = 0; i < n; i++){
        cin >> str1[i];
    }
    for(int i = 0; i < n; i++){
        str2[i] = str1[i];
    }
    for(int i = 0; i < n; i++){
        reverse(str1[i].begin(), str1[i].end());
    }
    for(int i = 0; i < n; i++){
        cout << check(str1[i], str2[i]) << endl;
    }
    return 0;
}
```

## Palin drom

```
1 #include <stdio.h>
2 #include <string.h>
3
4 int main(){
5     char str[201], rev[201];
6     int T ;
7     scanf("%d", &T);
8     for(int m = 0; m < T; m++){
9         scanf("%s", str);
10
11         int n = strlen(str), j = 0;
12         for(int i = n - 1; i >= 0; i--){
13             rev[j] = str[i];
14             j++;
15         }
16
17         rev[j] = '\0';
18         (strcmp(str, rev) == 0) ? printf("Yes\n") : printf("No\n");
19     }
20
21     return 0;
22 }
```

**Anagram Words**

Anagram word เป็นคำ 2 คำที่ประกอบด้วยตัวอักษรเหมือนกัน แต่มีลำดับการเรียงอักขระในสตริงแตกต่างกันเช่น LISTEN กับ SILENT และ TRIANGLE กับ INTEGRAL เป็นต้น

จงเขียนโปรแกรมเพื่อตรวจสอบคำสองคำว่าเป็น anagram word หรือไม่

ข้อมูลนำเข้า      บรรทัดที่ 1 จำนวนชุดทดสอบ T โดยที่  $2 \leq T \leq 10$

บรรทัดที่ 2 ถึง T + 1 เป็นคำสองคำเว้นวรรคด้วยช่องว่าง 1 ช่อง โดยที่ความยาวของคำแทนด้วย n และ  $2 \leq n \leq 200$

ข้อมูลนำออก      ผลการตรวจสอบ Anagram words จำนวน T บรรทัด

ตัวอย่าง

| Input             | Output |
|-------------------|--------|
| 2                 | Yes    |
| LISTEN SILENT     | Yes    |
| TRIANGLE INTEGRAL |        |
| 3                 | No     |
| test teas         | Yes    |
| tar rat           | Yes    |
| elbow below       |        |

**Compression**

จงเขียนโปรแกรมเพื่อบีบอัดข้อความให้สั้นลง โดยใช้จำนวนแทนอักขระที่ปรากฏซ้ำกันในข้อความ

ข้อมูลนำเข้า      ชุดอักขระประกอบด้วยอักขระ A-Z

ข้อมูลนำออก      แสดงข้อความที่ผ่านกระบวนการบีบอัด

ตัวอย่าง

| Input           | Output     |
|-----------------|------------|
| AAABBBBDDDDDDDD | 3A4B7D     |
| XXYYYYZZZAAAABB | 3X4Y3Z4A2B |
| ABC             | ABC        |

## ! Anagram Word !

```
1 #include <iostream>
2 #include <bits/stdc++.h>
3 #include <string>
4
5 using namespace std;
6
7 string check(string s1, string s2){
8
9     sort(s1.begin(), s1.end());
10    sort(s2.begin(), s2.end());
11
12    if(s1 == s2){
13        return "Yes";
14    }else{
15        return "No";
16    }
17 }
18
19
20 int main(){
21     int n;
22     cin >> n;
23     string w[n][2];
24
25     for(int i = 0; i < n; i++){
26         cin >> w[i][0] >> w[i][1];
27     }
28
29
30     for(int i = 0; i < n; i++){
31         cout << check(w[i][0], w[i][1]) << endl;
32     }
33
34     return 0;
35 }
```



```

1 #include <iostream>
2 #include <string>
3 using namespace std;
4
5 int main(){
6
7     string s;
8     cin >> s;
9     char c = s[0];
10    int n = 0;
11
12    for(int i = 0; i < s.size(); i++){
13
14        if(c == s[i]){
15            n++;
16        }else{
17
18            if(n > 1){
19                cout << n << c;
20
21                c = s[i];
22                n = 1;
23
24            }else{
25                cout << c;
26
27                c = s[i];
28                n = 1;
29            }
30        }
31    }
32
33    if(n > 1){
34        cout << n << c;
35    }else{
36        cout << c;
37    }
38
39    return 0;
40
41 }

```



**Virus Detection**

จงเขียนโปรแกรมเพื่อตรวจสอบคำว่า virus ที่ปรากฏในไฟล์ โดยการปรากฏของคำว่า virus ปรากฏได้ 2 แบบ คือ

แบบที่ 1 ไฟล์มีคำว่า virus ปรากฏอยู่ในบรรทัดใดบรรทัดหนึ่งของไฟล์ เช่น ab12\$**virus**23

แบบที่ 2 ไฟล์มีคำว่า virus ปรากฏอยู่ในบรรทัดใดบรรทัดหนึ่งของไฟล์ แต่เป็นการปรากฏแบบที่อักขระของคำว่า virus ไม่อยู่ในตำแหน่งที่ต่อเนื่อง เช่น alirusv2vriirc\$u5#us9

ข้อมูลนำเข้า บรรทัดที่ 1 จำนวนบรรทัด แทนด้วย  $T$  โดยที่  $2 \leq T \leq 100$

บรรทัดที่ 2 ถึง  $T + 1$  เป็นข้อความที่ปรากฏในบรรทัดที่  $T$  โดยที่แต่ละบรรทัดข้อความยาวไม่เกิน 80 อักขระ

ข้อมูลนำออก หมายเลขบรรทัดที่มีคำว่า virus ปรากฏอยู่แบบที่ 1 หรือแบบที่ 2

กำหนดให้ หมายเลขบรรทัดเริ่มที่ 1

หมายเหตุ ชุดทดสอบทุกชุดมีอย่างน้อย 1 บรรทัดที่คำว่า virus

ตัวอย่าง

| Input   | Output      |
|---|-------------|
| 4<br>dsffdvi5wqdhdr<br>vvvvirus23e4<br>@#1vviwdscr32uu2ss<br>%%gfsd   | 2<br>3      |
| 6<br>Itisveryinvigoratingtorunaroundtheskyscraper.<br>This line does not contain the v*i*r*u*s string.<br>) (^*&V#@\$(i*)*)&R879078u*)*)*S<br>Nowisthetimeforallhackersandvvvviirrruuusssswriterstocease<br>fajlcvjioryweoprhavkahyiesdjvoefhbg<br>piroutrgbsfoguofgnktnhnskhkhkhkhkhbfskklhfeikbmadfgiyewt | 1<br>2<br>4 |

**Three Single Number**

จงเขียนโปรแกรมเพื่อค้นหาชุดของเลขโดด 3 ตัวที่ไม่ซ้ำและเรียงลำดับจากน้อยไปมาก โดยมีเงื่อนไขว่าผลรวมของเลขโดดทั้งสามนั้นมีค่ามากกว่า 10 และให้พิมพ์ผลลัพธ์ตามลำดับที่พบ หากพบชุดของตัวเลขที่ซ้ำกันให้พิมพ์แค่ครั้งเดียว และหากไม่พบชุดของตัวเลขตามเงื่อนไขที่กำหนดให้พิมพ์ข้อความ NONE

ข้อมูลนำเข้า ชุดอักขระ ประกอบด้วยเลข 0-9

ข้อมูลนำออก ชุดอักขระประกอบด้วยตัวเลข 0-9 ความยาวขนาด 3 และเป็นไปตามเงื่อนไขที่กำหนด และจะไม่พิมพ์ชุดอักขระที่เป็นตามเงื่อนไขซ้ำ

ตัวอย่าง

| Input                | Output      |
|----------------------|-------------|
| 1246858594124665     | 246 468     |
| 124680585941666      | 246 468 058 |
| 12468058594246666058 | 246 468 058 |
| 235124               | NONE        |

## นับพยางค์

พยางค์คือหน่วยของเสียงในการพูด เช่นคำว่า computer เมื่ออ่านว่า คอ-ม-พิว-เตอ ก็จะมี 3 พยางค์เป็นต้น

ปกติแล้วการนับพยางค์ให้ถูกต้องทำได้ยาก แต่เราจะพิจารณาวิธีการนับพยางค์แบบง่าย ๆ

ในภาษาอังกฤษมีสระอยู่ทั้งสิ้น 5 ตัวคือ a e i o และ u เราจะนับให้พยางค์เท่ากับจำนวนครั้งของสระที่ปรากฏติดกันในคำ ยกตัวอย่างเช่น

- คำว่า computer มีสระปรากฏติดกันทั้งสิ้น 3 ครั้ง จะนับให้มี 3 พยางค์
- คำว่า seal มีสระปรากฏติดกันทั้งสิ้น 1 ครั้ง จะนับให้มี 1 พยางค์
- คำว่า deareioviewreiura มีสระปรากฏติดกันทั้งสิ้น 5 ครั้ง จะนับให้มี 5 พยางค์

## งานของคุณ

เขียนโปรแกรมรับรายการของคำภาษาอังกฤษและคำนวณจำนวนพยางค์ตามวิธีที่ระบุข้างต้น

## ข้อมูลนำเข้า

บรรทัดแรกระบุจำนวนเต็ม N แทนจำนวนคำที่ให้คำนวณ ( $1 \leq N \leq 10$ ) จากนั้นอีก N บรรทัดจะระบุคำคำละหนึ่งบรรทัด แต่ละคำจะประกอบไปด้วยตัวอักษรภาษาอังกฤษพิมพ์เล็ก และมีความยาวไม่เกิน 30 ตัวอักษร

## ข้อมูลส่งออก

มี N บรรทัด แต่ละบรรทัดเป็นจำนวนเต็มหนึ่งจำนวน ระบุจำนวนพยางค์ของคำแต่ละคำในรายการ

## ตัวอย่างข้อมูลนำเข้าและส่งออก

| ตัวอย่างข้อมูลนำเข้า | ตัวอย่างข้อมูลส่งออก |
|----------------------|----------------------|
| 4                    | 2                    |
| hello                | 2                    |
| already              | 3                    |
| computer             | 1                    |
| seal                 |                      |

|                          |  |                             |
|--------------------------|--|-----------------------------|
| โจทย์แข่งขัน TOI.A       |  | หน้าที่ 2 จากทั้งหมด 2 หน้า |
| รอบประจำเดือนเมษายน 2553 |  | ชื่อโจทย์: <b>syllables</b> |

ข้อจำกัดของโปรแกรม

โปรแกรมของคุณต้องทำงานภายในเวลา 1 วินาที และใช้หน่วยความจำไม่เกิน 32 MB



```

1 #include <iostream>
2 #include <bits/stdc++.h>
3 #include <string>
4 using namespace std;
5
6 bool check(char x){
7
8     string a = "aeiou";
9     for(int i = 0; i < a.size(); i++){
10         if(x == a[i]){
11             return false;
12         }
13     }
14
15     return true;
16 }

```

```

17
18 int main(){
19     int n;
20     cin >> n;
21
22     string words[n];
23     string x = "aeiou";
24
25     int a = 0;
26
27     for(int i = 0; i < n; i++){
28         cin >> words[i];
29     }
30
31     for(int k = 0; k < n; k++){
32
33         string s = words[k];

```

```

34
35         for(int i = 0; i < s.size(); i++){
36
37             for(int j = 0; j < x.size(); j++){
38
39                 if(s[i] == x[j] && check(s[i + 1])){
40                     a++;
41                 }
42             }
43         }
44
45         cout << a << endl;
46         a = 0;
47     }
48
49     return 0;
50 }

```

**1000 : Word Chain****Problem type** : Batch**Time limit** : 1.0 second(s)**Memory limit** : 64 megabyte(s)

**โซ่คำ** คือลำดับของคำที่มีจำนวนอักขระเท่ากันและแต่ละคำที่มีลำดับติดกันจะต้องมีตำแหน่งที่มีตัวอักขระต่างกันไม่เกินสองตำแหน่ง เช่น HEAD และ HEAP จะต่างกันตำแหน่งเดียวคือ D และ P ในตำแหน่งตัวอักขระที่ 4 ของคำ ในขณะที่ REAR กับ BAER จะมีตำแหน่งต่างกัน 3 ตำแหน่ง คือ ตำแหน่งที่ 1 (R กับ B) ตำแหน่งที่ 2 (E และ A) และ ตำแหน่งที่ 3 (A และ E)

**ตัวอย่างของโซ่คำที่ต่อเนื่องได้แก่** HEAD HEAP LEAP TEAR REAR และ EGG EAG GAE GAP TAP TIN

**ตัวอย่างของโซ่คำที่ขาดได้แก่** LEAP TEAR REAR BAER BAET BEEP ซึ่งจะขาดที่ คำว่า BAER

ให้ชุดของโซ่คำมาชุดหนึ่ง จงเขียนโปรแกรมเพื่อหาคำสุดท้ายในโซ่คำ ก่อนที่โซ่คำจะขาด

**ข้อมูลนำเข้า**

**บรรทัดแรก** เก็บจำนวนเต็ม L แทนจำนวนตัวอักขระของแต่ละคำ โดยที่  $3 \leq L \leq 1000$

**บรรทัดที่สอง** เก็บจำนวนเต็ม N แทนจำนวนคำทั้งหมดในแฟ้มข้อมูล โดยที่  $1 \leq N \leq 30000$

**บรรทัดที่ 3..(N +2)** เก็บลำดับของคำที่มีจำนวนตัวอักขระ L ตัว แต่ละบรรทัดเก็บคำที่เขียนด้วยตัวอักษร ( 'A' ถึง 'Z' ) ที่เป็นตัวพิมพ์ใหญ่

**ข้อมูลส่งออก**

แฟ้มผลลัพธ์ มี 1 บรรทัด เก็บคำสุดท้ายของโซ่คำชุดแรก

**ที่มา:** การแข่งขันคอมพิวเตอร์โอลิมปิก สอวน. ครั้งที่ 1 มหาวิทยาลัยเกษตรศาสตร์

| ตัวอย่างข้อมูลนำเข้า  | ตัวอย่างข้อมูลส่งออก |
|---|----------------------|
| 4<br>12<br>HEAD<br>HEAP<br>LEAP<br>TEAR<br>REAR<br>BAER<br>BAET<br>BEEP<br>JEEP<br>JOIP<br>JEIP<br>AEIO | REAR                 |

**ความช่วยเหลือ:** ไม่มีคำใบ้สำหรับปัญหานี้

```

1 #include <iostream>
2 #include <bits/stdc++.h>
3 #include <string>
4 using namespace std;
5
6 int main(){
7     int l; //4
8     int n; //12
9
10    cin >> l >> n;
11
12    string s[n];
13
14    for(int i = 0; i < n; i++){
15        cin >> s[i];
16    }
17
18
19
20    int a = 0;
21    string w = s[0];
22
23    for(int i = 0; i < n; i++){
24
25        for(int j = 0; j < 4; j++){
26
27            if(w[j] == s[i][j]){
28
29                a++;
30
31            }
32        }
33
34        if(a < 2){
35            cout << w;
36            break;
37        }else{
38            a = 0;
39            w = s[i];
40        }
41
42    }
43
44
45    return 0;
46 }
47
48
49
50 // string name[2] = {"kamal", "baha"};
51 // for(int i = 0; i < name[1].size(); i++){
52 //     cout << name[1][i] << endl;
53 // }

```

S(10)

**0004 : Character Checker**  
**Problem type : Batch**  
**Time limit : 1.0 second(s)**  
**Memory limit : 64 megabyte(s)**

**โจทย์**

ให้นักเรียนตรวจสอบว่า สายอักขระที่โจทย์กำหนดให้ เป็นตัวอักษรพิมพ์เล็กทั้งหมด หรือตัวอักษรพิมพ์ใหญ่ทั้งหมด หรือมีทั้งตัวอักษรพิมพ์ใหญ่และพิมพ์เล็กผสมกันอยู่

**ข้อมูลนำเข้า**

บรรทัดแรกเพียงบรรทัดเดียว สายอักขระ ที่ประกอบไปด้วยตัวอักษร 'a' - 'z' หรือ 'A' - 'Z' เท่านั้น ความยาวไม่เกิน 10,000 ตัวอักษร

**ข้อมูลส่งออก**

ให้พิมพ์คำตอบตามเงื่อนไขต่อไปนี้

- ถ้าสายอักขระประกอบด้วยตัวพิมพ์ใหญ่ทั้งหมด ให้พิมพ์ว่า "All Capital Letter" โดยไม่ต้องมีเครื่องหมาย ""
- ถ้าสายอักขระประกอบด้วยตัวพิมพ์เล็กทั้งหมด ให้พิมพ์ว่า "All Small Letter" โดยไม่ต้องมีเครื่องหมาย ""
- ถ้าสายอักขระประกอบด้วยตัวอักษรพิมพ์เล็กและพิมพ์ใหญ่ผสมกันไป ให้พิมพ์ว่า "Mix" โดยไม่ต้องมีเครื่องหมาย ""

ที่มา: Programming.in.th (Northern\_series)

| ตัวอย่างข้อมูลนำเข้า | ตัวอย่างข้อมูลส่งออก |
|----------------------|----------------------|
| ABCDEFGH             | All Capital Letter   |
| abcdefgh             | All Small Letter     |
| AbCdEfG              | Mix                  |

C

อยู่ใน Programming.in.th  
C++

```

1 #include <stdio.h>
2 #include <string.h>
3
4 int main(){
5     char str[10000];
6     gets(str);
7     int n, a = 0, b = 0;
8     n = strlen(str);
9
10    for(int i = 0; i < n; i++){
11        if((str[i] >= 'A') && (str[i] <= 'Z')){
12            a++;
13        }else if((str[i] >= 'a') && (str[i] <= 'z')){
14            b++;
15        }
16    }
17
18    if(a > 0 && b == 0){
19        printf("All Capital Letter");
20    }else if(a == 0 && b > 0){
21        printf("All Small Letter");
22    }else{
23        printf("Mix");
24    }
25
26    return 0;
27 }

```

```

1 #include <bits/stdc++.h>
2 using namespace std;
3
4 int main(){
5     string s;
6     cin >> s;
7     int a = 0, b = 0;
8
9     for(int i = 0; i < s.size(); i++){
10        if(s[i] >= 'A' && s[i] <= 'Z'){
11            a++;
12        }else if(s[i] >= 'a' && s[i] <= 'z'){
13            b++;
14        }
15    }
16
17    if(a > 0 && b == 0){
18        cout << "All capital Letter";
19    }else if(a == 0 && b > 0){
20        cout << "All Small Letter";
21    }else{
22        cout << "Mix";
23    }
24    return 0;
25 }

```